Trial Projects - Dec 2020

SQL analysis

Description

The CWE by MITRE includes common SQL flaws and anti-patterns.

Some examples:

- CWE-1049: Excessive Data Query Operations in a Large Data Table - Too many joins, too many sub-queries.
- CWE-1050: Excessive Platform Resource Consumption within a Loop - Running SQL in a loop, or "N+1 query" problem (see my blog post on this subject)
- CWE-1067: Excessive Execution of Sequential Searches of Data Resource - Similar to 1050.
- CWE-1084: Invokable Control Element with Excessive File or Data Access Operations - Too many queries in the same operation. May also be an indication of an N+1 query issue.
- CWE-1094: Excessive Index Range Scan for a Data Resource - Un-indexed join.
- CWE-1099: Inconsistent Naming Conventions for Identifiers - Un-expected or unconventional column name used in a join (e.g. with a primary key).

Detect these flaws and anti-patterns in the provided dataset. You can simulate the database metadata, for example information about primary keys, available indexes, etc.

Tips

A good Ruby library for analyzing SQL is pg_query. This will give you a parse tree of PostgreSQL-compatible queries. MySQL queries can generally be made to work with a bit pre-processing.

Deliverable

A program which takes a directory of text files containing SQL queries as input, and produces:

1) A list of detected flaws and other issues.
2) [optional] A schema description which is inferred from the SQL. The schema includes tables, foreign keys, and columns. No database metadata is used to build the inferred schema; it's generated strictly from the SQL.

Data set

https://drive.google.com/file/d/1HwlXeiSJPcp5Ot4hp_ijk9Z1RartZkFH/view?usp=sharing

AppMap events optimization

Description

AppLand clients generate files in the appmap.json format. Sometimes, the appmap files generated by the client contain a lot of "noise"; for example, getters and setters. Users can configure an exclusion list to filter these out; but often they don't. So the exercise is to automatically remove the "non interesting" events (e.g. by some information theory or statistical methods), compacting the files and improving their usefulness.

Deliverable

A program which takes an appmap.json file as input, and emits an appmap.json file with "noisy" events removed. The program may take user-configurable options (such as the degree of desired size reduction, target output size, or statistical parameters), or it may just create a "best guess" output.

Tips

To get a quick idea of the "shape" of an appmap file, you can use the appland stats command.

Data set

https://drive.google.com/file/d/14XPIA19CQi59aOpjZCKKkJzpERT2KpHw/view?usp=sharing